```
?  show files
[File 60] ANTE: Abstracts in New Tech & Engineer 1966-2008/Nov
(c) 2008 CSA. All rights reserved.
[File 56] Computer and Information Systems Abstracts 1966-2008/Nov
(c) 2008 CSA. All rights reserved.
[File 35] Dissertation Abs Online 1861-2008/Feb
(c) 2008 ProQuest Info&Learning. All rights reserved.
[File 8] Ei Compendex(R) 1884-2008/Nov W5
(c) 2008 Elsevier Eng. Info. Inc. All rights reserved.
[File 266] FEDRIP 2008/Oct
Comp & dist by NTIS, Intl Copyright All Rights Res. All rights reserved.
[File 583] Gale Group Globalbase(TM) 1986-2002/Dec 13
(c) 2002 Gale/Cengage. All rights reserved.
*File 583: This file is no longer updating as of 12-13-2002.
[File 65] Inside Conferences 1993-2008/Dec 17
(c) 2008 BLDSC all rts. reserv. All rights reserved.
[File 2] INSPEC 1898-2008/Nov W3
(c) 2008 Institution of Electrical Engineers. All rights reserved.
[File 6] NTIS 1964-2008/Dec W4
(c) 2008 NTIS, Intl Cpyrght All Rights Res. All rights reserved.
[File 144] Pascal 1973-2008/Dec W2
(c) 2008 INIST/CNRS. All rights reserved.
[File 34] SciSearch(R) Cited Ref Sci 1990-2008/Dec W2
(c) 2008 The Thomson Corp. All rights reserved.
[File 434] SciSearch(R) Cited Ref Sci 1974-1989/Dec
(c) 2006 The Thomson Corp. All rights reserved.
[File 256] TecInfoSource 82-2008/Jul
(c) 2008 Info.Sources Inc. All rights reserved.
[File 95] TEME-Technology & Management 1989-2008/Nov W5
(c) 2008 FIZ TECHNIK. All rights reserved.
[File 99] Wilson Appl. Sci & Tech Abs 1983-2008/Oct
(c) 2008 The HW Wilson Co. All rights reserved.


: d s
Set     Items   Description
S1        560   S (KERNEL(3N)(EMULAT? OR SIMULATOR? ? OR VIRTUAL))
S2     274975   S ( "NON" ()NATIVE OR NONNATIVE OR NATIVE)
S3       5237   S (CONVERT? OR CONVERSION? ? OR CHANGE?? OR CHANGING OR TRANSLAT? OR CHANG? OR
ALTER? OR ALTERATION? ? OR MODIF? OR TRANSFORM? OR REPLAC??? OR SUBSTITUT????)(2N)KERNEL? ?
S4          5   S S3(20N)S2
S5          0   S S4(20N)S1
S6         13   S S1 AND S3
S7          8   RD  (unique items)
S8          0   S S1 AND (AU=( BOND, B? OR BOND B OR KHALID, A? OR KHALID A? OR KHALID, S? OR KHALID S?))

?
```

Subject summary

? 1 /5,k/all
7/5,K/1 (Item 1 from file: 56) Links
Fulltext available through: STIC Full Text Retrieval Options
Computer and Information Systems Abstracts

0000614878 IP Accession No: 200702-90-011194
The virtual network system
Casado, Martin; McKeown, Nick
ACM SIGCSE Bulletin , v 37 , n 1 , p 76-80 , 2005
Publication Date: 2005
Publisher: Association for Computing Machinery, Inc. , One Astor Plaza, 1515 Broadway , New York , NY , 10036-5701
Country Of Publication: USA
Publisher Url: http://www.acm.org/
Publisher Email: SIGS@acm.org
Document Type: Electronic Journal Article
Record Type: Abstract
Language: English
ISSN: 0097-8418
DOI: 10.1145/1047124.1047383
File Segment: Computer & Information Systems Abstracts
Abstract:
The goal of our work is to give students a hands-on experience designing, deploying and debugging parts of the Internet infrastructure, such as an Internet router that routes real network traffic, or a security firewall. To do so normally requires that the students have access to snoop and generate raw network traffic, which is a risk to privacy and security. And it normally requires each student to have a dedicated computer, and to modify the kernel. The Virtual Network System (VNS) is a teaching tool designed for undergraduate and graduate networking courses. With VNS, each student can build a router (or any packet-processing device) in user-space, in their own private, protected topology, and process real Internet traffic. VNS has been used by over 500 students at Stanford and remotely from other universities. This paper describes the VNS tool, and our experiences using it in the classroom.
Descriptors: Internet; Traffic flow; Traffic engineering; Routers; Tools; Networks; Construction; Virtual networks; Computer information security; Firewalls; Teaching; Risk; Infrastructure; Computer networks; Education; Classrooms; Debugging; Routing (telecommunications); Privacy; Graduates; Kernels; Topology; Courses
Subj Catg: 90, Computing Milieux (General)
Abstract:
...and security. And it normally requires each student to have a dedicated computer, and to modify the kernel. The Virtual Network System (VNS) is a teaching tool designed for undergraduate and graduate networking courses. With...

7/5,K/2 (Item 2 from file: 56) Links
Computer and Information Systems Abstracts

0000392144 IP Accession No: 200406-22-0095
The virtual processor interface: Linux kernel support for user-level thread systems.
Benson, G D; Butner, M; Padden, Sh; Fedosov, A University of California (San Francisco)
Author Email: benson@cs.usfca.edu
Pages: 681-687
Publication Date: 2003
Publisher: International Association of Science and Technology for Development , #80, 4500 - 16 Ave NW , Calgary, Alberta , T3B 0M6
Country Of Publication: Canada
Publisher Url: http://www.iasted.com
Publisher Email: Calgary@iasted.org
Conference:
Fifteenth IASTED International Conference on Parallel and Distributed Computing and Systems , Marina del Ray, CA , USA , 3-5 Nov. 2003
Document Type: Conference Paper
Record Type: Abstract
Language: English
ISBN: 0-88986-392-X
Notes: Graphs
No. Of Refs.: 19
File Segment: Computer & Information Systems Abstracts
Abstract:
Despite an increasing need for thread support in language run-time systems and parallel libraries such as in Java and OpenMP, there is limited support for custom, multiprocessor capable, user-level thread systems in the Linux kernel. To address this lack of support we have developed the virtual processor interface (VPI) for Linux. Our VPI implementation consists of a small set of kernel modifications and new system calls combined with a small user-level library that provide

an interface that can be used to build thread systems. VPI uses a form of scheduler activations so that user-level thread systems can have complete control over the amount of parallelism for an application and the scheduling of threads onto processors. In addition, VPI allows user-level thread systems to schedule new threads in the presence of blocking system calls and page faults. This paper describes VPI and our implementation. We have implemented a complete thread system using VPI, called VPIthreads, and compare its performance to that of current user-level and kernel-level thread systems. Our initial results show the VPI-based thread systems can perform better than current production thread systems.
Descriptors: Systems; Microprocessors; Libraries; Production; Activation; Scheduling; Computer networks; Programming languages; Performance; Schedules
Subj Catg: 22, Processor Architectures and Process Management
The virtual processor interface: Linux kernel support for user-level thread systems.
Abstract:
...virtual processor interface (VPI) for Linux. Our VPI implementation consists of a small set of kernel modifications and new system calls combined with a small user-level library that provide an interface...

7/5,K/3 (Item 1 from file: 35) Links
Dissertation Abs Online

01831865 ORDER NO: AADAA-I3012433
Fine-grained dynamic instrumentation of commodity operating system kernels
Author: Tamches, Ariel Meir
Degree: Ph.D.
Year: 2001
Corporate Source/Institution: The University of Wisconsin - Madison (0262)
Supervisor: Barton P. Miller
Source: Volume 6204B of Dissertations Abstracts International.
PAGE 1945 . 213 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984
ISBN: 0-493-22771-7

Operating system kernel code is generally immutable. This trend is unfortunate, because a kernel that can insert (and later remove) code at run-time has many uses, including performance measurement, debugging, code coverage, run-time installation of patches, and run-time optimizations. The research in this dissertation investigates dynamic (run-time) kernel instrumentation and its applications in the areas of kernel profiling and code evolution. We have implemented dynamic kernel instrumentation, a kernel performance monitor, and a run-time kernel optimizer in a system called <italic>KernInst</italic>.

The first component of this dissertation investigates <italic>fine-grained dynamic kernel instrumentation</italic>, a technology to dynamically modify kernel code. We have designed two primitives for run-time kernel code modification, <italic> splicing</italic>, which inserts instrumentation code, and <italic>code replacement </italic>, which replaces a function's code. A part of the KernInst system called <italic>kerninstd</italic> implements fine-grained dynamic instrumentation for Solaris UltraSPARC kernels.

The second component of this dissertation is the collection of techniques and algorithms for using dynamic instrumentation to obtain kernel performance information. The first techniques are the design and implementation of efficient instrumentation code to obtain counts, elapsed times, and virtual times of kernel code. This dissertation also presents a means to effectively calculate an estimate of kernel control flow graph counts from basic block counts. These techniques and algorithms are embodied in a kernel performance tool called <italic> kperfmon</italic>. A case study describes how kperfmon helped to understand and improve the performance of a Web proxy server.

The final component of this dissertation introduces the concept of <italic> evolving code</italic> in a commodity operating system. An evolving kernel changes it code at run-time, in response to the measured environment. KernInst provides a technological infrastructure that enables commodity kernels to evolve. As a proof of concept, we describe an automated kernel run-time version of the code positioning I-cache optimization. We have applied run-time code positioning to the TCP read-side stream processing routine <tfont>tcp rput data </tfont>. Code positioning reduces the time that <tfont>tcp rput data</tfont> waits for I-cache misses by 35%, reduces its execution time by 17%, and improves its instructions per cycle by 36%.

...of this dissertation investigates <italic>fine-grained dynamic kernel instrumentation</italic>, a technology to dynamically modify kernel code. We have designed two primitives for run-time kernel code modification, <italic> splicing</italic>, which inserts instrumentation code, and <italic>code replacement </italic>, which replaces a... ...are the design and implementation of efficient instrumentation code to obtain counts, elapsed times, and virtual times of kernel code. This dissertation also presents a means to effectively calculate an estimate of kernel control... ...introduces the concept of <italic> evolving code</italic> in a commodity operating system. An evolving kernel changes it code at run-time, in response to the measured environment. KernInst provides a technological...

7/5,K/4 (Item 1 from file: 8) Links
Ei Compendex(R)

0018436117 E.I. COMPENDEX No: 20082811359169
IMUNES based distributed network emulator
Issue Title: SoftCOM 2006 - International Conference on Software, Telecommunications andComputer Networks
Puljiz, Z.; Mikuc, M.
Corresp. Author/Affil: Puljiz, Z.: Faculty of Electrical Engineering and Computing, Department of Telecommunications, Zagreb, Croatia

Corresp. Author email: zrinka.puljiz@fer.hr
Author email: miljenko.mikuc@fer.hr
Conference Title: SoftCOM 2006 - International Conference on Software, Telecommunications andComputer Networks
Conference Location: Split-Dubrovnik Croatia Conference Date: 20060929-20061001
Sponsor: IEEE Communications Society (COMSOC); Minist. Sea, Tourism, Transport Dev. Republic of Croatia; Croatian Academy of Engineering; Croatian Academy of Sciences and Arts
E.I. Conference No.: 72416
SoftCOM 2006 - International Conference on Software, Telecommunications andComputer Networks (SoftCOM - Int. Conf. Softw., Telecommun. Comp. Netw.) (United States) 2006 , IEEE 06EX1452 (198-203)
Publication Date: 20060101
Publisher: Inst. of Elec. and Elec. Eng. Computer Society
ISBN: 953611487X 9789536114870
Item Identifier (DOI): 10.1109/SOFTCOM.2006.329743
Article Number: 4129897
Document Type: Conference Paper; Conference Proceeding Record Type: Abstract
Language: English Summary Language: English
Number of References: 14
In this paper we describe a new version of our distributed network emulator that extends an existing kernel level emulator called IMUNES. IMUNES is based on a lightweight virtual machine concept and performs zero copying when packets traverse through the emulated topology. It works on a modified FreeBSD kernel and enables emulated nodes to use the standard UNIX applications. The main strengths of this tool are scalability, performance and high fidelity. We are developing a distributed network simulation to further increase the scalability by allowing parts of emulation to be deployed across a peer-to-peer emulator cluster. The decentralized management of the emulator cluster improves availability and robustness of the system. We provide support for a multi-user and multi-experiment environment to maximize the benefit from newly increased resources.
Descriptors: Ad hoc networks; Computer software; Scalability; Standards; Telecommunication systems; Virtual reality; *Simulators
Identifiers: Decentralized management; Distributed network simulation; Distributed networks; FreeBSD; High Fidelity; international conferences; Multi users; Peer-to-peer (p2p); Virtual machine (VM)
Classification Codes:
722.3 (Data Communication, Equipment & Techniques)
902.2 (Codes & Standards)
621 (Nuclear Reactors)
716 (Electronic Equipment, Radar, Radio & Television)
723 (Computer Software, Data Handling & Applications)
961 (Systems Science)
...paper we describe a new version of our distributed network emulator that extends an existing kernel level emulator called IMUNES. IMUNES is based on a lightweight virtual machine concept and performs zero copying when packets traverse through the emulated topology. It works on a modified FreeBSD kernel and enables emulated nodes to use the standard UNIX applications. The main strengths of this tool are scalability...
Descriptors:

7/5,K/5 (Item 2 from file: 8) Links
Ei Compendex(R)

0016216583 E.I. COMPENDEX No: 2004488475160
Complete or fast reference trace collection for simulating multiprogrammed workloads: Choose one
Issue Title: SIGMETRICS 2004/Performance 2004: Joint International Conference on Measurement and Modeling of Computer Systems
Kaplan, Scott F.
Corresp. Author/Affil: Kaplan, S.F.: Department of Mathematics, Amherst College, Amherst, MA 01002-5000, United States
Corresp. Author email: sfkaplan@cs.amherst.edu
Conference Title: SIGMETRICS 2004/Performance 2004: Joint International Conference on Measurement and Modeling of Computer Systems
Conference Location: New York, NY United States Conference Date: 20040612-20040616
Sponsor: ACM, SIGMETRICS; IFIP Working Group 7.3
E.I. Conference No.: 63837
Performance Evaluation Review (Perform Eval Rev) (United States) 2004 32/1 (420-421)
Publication Date: 20041124
Publisher: Association for Computing Machinery
CODEN: PERED ISSN: 0163-5999
Document Type: Conference Paper; Conference Proceeding Record Type: Abstract
Treatment: T; (Theoretical)
Language: English Summary Language: English
Number of References: 8
The various aspects of two new collectors, Laplace and kVMTrace, that log the kernel-level information needed to drive multiprogrammed simulations, are discussed. There are two components to Laplace, namely, a modified machine simulator and a modified kernel. There is only one component to kVMTrace, the modified kernel, which emits both the reference and kernel event traces, where each contains the same basic information as with Laplace. The post-processor

reconciles the information from the two streams and them emits a set of traces, one per thread, that can be used as the input for a multiprogrammed simulation.
Descriptors: Computer simulation; Data storage equipment; Encoding (symbols); Laplace transforms; Mapping; Queueing networks; *Multiprogramming
Identifiers: Memory simulations; Multiprogrammed workloads; Reference trace collection; Trace-driven simulation
Classification Codes:
405.3 (Surveying)
722.1 (Data Storage, Equipment & Techniques)
723.1 (Computer Programming)
723.2 (Data Processing)
723.5 (Computer Applications)
921.3 (Mathematical Transformations)
...drive multiprogrammed simulations, are discussed. There are two components to Laplace, namely, a modified machine simulator and a modified kernel. There is only one component to kVMTrace, the modified kernel, which emits both the reference and kernel event traces, where each contains the same basic...
Descriptors:

7/5,K/6 (Item 3 from file: 8) Links
Ei Compendex(R)

0014168966 E.I. COMPENDEX No: 1998424349149
Microscopic simulator of traffic flow
Duan, Jin-yu; Yang, Pei-kun
Corresp. Author/Affil: Duan, Jin-yu: Tongji Univ, Shanghai, China
Editor(s): Yang, Z.; Wang, K.C.P.; Mao, B.
Conference Title: Proceedings of the 1998 Conference on Traffic and Transportation Studies, ICTTS
Conference Location: Beijing, China Conference Date: 19980727-19980729
Sponsor: ASCE
E.I. Conference No.: 48832
Proceedings of the Conference on Traffic and Transportation Studies, ICTTS (Proc Conf Traffic Transport Stud ICTTS) 1998 (633-641)
Publication Date: 19980101
Publisher: ASCE
CODEN: 00310
Document Type: Conference Paper; Conference Proceeding Record Type: Abstract
Treatment: G: (General review)
Language: English Summary Language: English
Using object-oriented analyzing method, this paper interprets the problem of microscopic traffic simulation in the merging area of expressway mainline with on-ramp in a more understandable manner. The models of car following and lane changing, as the kernel of the simulator named as MicroSim, are discussed in detail. For the purpose of full practice, special efforts are laid on boundary circumstances handling in car following model. Based on the imitation of the intellectual pattern of a driver, a uniform mechanism is described. In this mechanism, expressway lane changing logic and ramp merging are presented. Additionally, some simulation results are introduced.
Descriptors: Computer simulation; Motor transportation; Object oriented programming; *Highway traffic control
Identifiers: Microscopic traffic simulation; Software package MicroSim
Classification Codes:
432.4 (Highway Traffic Control)
723.1 (Computer Programming)
723.5 (Computer Applications)
...with on-ramp in a more understandable manner. The models of car following and lane changing, as the kernel of the simulator named as MicroSim, are discussed in detail. For the purpose of full practice, special efforts...
Descriptors:

7/5,K/7 (Item 4 from file: 8) Links
Ei Compendex(R)

0013132514 E.I. COMPENDEX No: 1994031189638
Image transformation method for determining kernel motion positions in three dimensions
Yang, Y.; Schrock, M.D.
Corresp. Author/Affil: Yang, Y.: Univ of Wisconsin, River Falls, United States
Transactions of the American Society of Agricultural Engineers (Trans ASAE) 1993 36/4 (1229-1234)
Publication Date: 19931201
CODEN: TAAEA ISSN: 0001-2351
Document Type: Article; Journal Record Type: Abstract
Treatment: X; (Experimental)
Language: English Summary Language: English
Number of References: 10
An image transformation method for determining kernel coordinates in three dimensions was developed. Instead of using two cameras to recover the lost dimension caused by projecting a three-dimensional object onto a two-dimensional film plane, one camera (with the help of a mirror and strobe light) was used to take photos of discrete kernel motion trajectory

with two views - one real image and one virtual image from the mirror. The image transformations were derived to determine kernel motion positions in three dimensions on the basis of the real kernel images and virtual kernel images from the photos.
Descriptors: Cameras; Combines; Image processing; Mirrors; Photography; Three dimensional; *Grain (agricultural product)
Identifiers: Image transformation; Kernel motion positions; Strobe light
Classification Codes:
741.3 (Optical Devices & Systems)
742.1 (Photography)
821.1 (Agricultural Machinery & Equipment)
821.4 (Agricultural Products)
...derived to determine kernel motion positions in three dimensions on the basis of the real kernel images and virtual kernel images from the photos.
Descriptors:
Identifiers: Image transformation; Kernel motion positions; Strobe light

7/5,K/8 (Item 1 from file: 144) Links
Fulltext available through: STIC Full Text Retrieval Options
Pascal

12833962 PASCAL No.: 97-0053389
SoftFLASH : Analyzing the performance of clustered distributed virtual shared memory
Architectural support for programming languages and operating systems
ERLICHSON A; NUCKOLLS N; CHESSON G; HENNESSY J
Computer Systems Lab, Stanford University, Stanford, CA 94305, United States; Silicon Graphics Inc., 2011 North Shoreline Blvd., Mountain View, CA 94043, United States
Association for Computing Machinery, New York NY, United States.; IEEE Computer Society, United States.
ASPLOS-VII. International Conference, 7 (Cambridge, MA USA) 1996-10-01
Journal: Operating systems review, 1996, 30 (5) 210-220
ISSN: 0163-5980 CODEN: OSRED8 Availability: INIST-18399 ; 354000060859980190
No. of Refs.: 28 ref.
Document Type: P (Serial); C (Conference Proceedings) ; A (Analytic)
Country of Publication: United States
Language: English
One potentially attractive way to build large-scale shared-memory machines is to use small-scale to medium-scale shared-memory machines as clusters that are interconnected with an off-the-shelf network. To create a shared-memory programming environment across the clusters, it is possible to use a virtual shared-memory software layer. Because of the low latency and high bandwidth of the interconnect available within each cluster, there are clear advantages in making the clusters as large as possible. The critical question then becomes whether the latency and bandwidth of the top-level network and the software system are sufficient to support the communication demands generated by the clusters. To explore these questions, we have built an aggressive kernel implementation of a virtual shared-memory system using SGI multiprocessors and 100Mbyte/sec HIPPI interconnects. The system obtains speedups on 32 processors (four nodes, eight processors per node plus additional reserved protocol processors) that range from 6.9 on the communication-intensive FFT program to 21.6 on Ocean (both from the SPLASH 2 suite). In general, clustering is effective in reducing internode miss rates, but as the cluster size increases, increases in the remote latency, mostly due to increased TLB synchronization cost, offset the advantages. For communication-intensive applications, such as FFT, the overhead of sending out network requests, the limited network bandwidth, and the long network latency prevent the achievement of good performance. Overall, this approach still appears promising, but our results indicate that large low latency networks may be needed to make cluster-based virtual shared-memory machines broadly useful as large-scale shared-memory multiprocessors.
English Descriptors: Distributed system; Shared memory; Virtual memory; Aggregation; System architecture; RISC processor; Cache memory; UNIX system; Parallelism; Addressing; Synchronization; Computer network; System performance

French Descriptors: Systeme reparti; Memoire partagee; Memoire virtuelle; Agregation; Architecture systeme; Processeur RISC; Antememoire; Systeme UNIX; Parallelisme; Adressage; Synchronisation; Reseau ordinateur; Performance systeme; DMA; Relaxed memory model; Kernel; Dedicated processor; Translation lookaside buffer; Adress space
Classification Codes: 001D02B04; 001D02B07C; 001D02B06

... communication demands generated by the clusters. To explore these questions, we have built an aggressive kernel implementation of a virtual shared-memory system using SGI multiprocessors and 100Mbyte/sec HIPPI interconnects. The system obtains speedups...
...French Descriptors: RISC; Antememoire; Systeme UNIX; Parallelisme; Adressage; Synchronisation; Reseau ordinateur; Performance systeme; DMA; Relaxed memory model; Kernel; Dedicated processor; Translation lookaside buffer; Adress space

?